Programming Test - R&D team

Overview

You will now be asked to perform a practical task to allow us to evaluate your skills.

ATTENTION: The test is designed to be too long for most programmers.
Please concentrate only on what you think you can do best and deliver most completely!

The server side tasks are the important part. Please finish these tasks first.

Your evaluation will be factored by the following parameters:

1. Percentage of major functionality completed
2. Code style and correctness
3. Architectural integrity
4. Visual styling / Look and Feel

Description

You have been requested to create a web application that allows customers to view and edit a video rental store.

On client side code you should use HTML5, CSS3 and optionally LESS/SCSS/STYLUS or any other and Javascript with JQuery or Angular. You may add any client side libraries as needed.

On the server side you may use NodeJS (javascript) or PHP. You can add any framework, library or component you need.

You should use a MySQL DB to store the primary application data and any other DB as needed.

You may use any hosting environment as you see fit, any cloud vendor and any SAAS or PAAS
You may use local installed environment of your machine

Finally - to share your code with me please send me link to download it
It can be in GITHUB or simple FTP location to download

The DB

You should use the Sakila sample database.
You will need to provision a MySQL DB server
If you are using local installed environment you can install also MySQL DB server
The link also allows you to download a DUMP of the DB so you can import it into the server you provision.

Main test - Server Side Task

All API should be RESTful
Latency should be kept to minimum.

1. Search API
    a. Will allow to search films by: (any combination)
        i. Title
        ii. Description
        iii. Category Name
        iv. Actor name
        v. Language Name
    b. Each search will be recorded into a table ('search_log') with the following schema:
        i. Timestamp
        ii. Search types (actor,title,desc,category…)
        iii. Search values
        iv. Result count
    c. Each search result should include following info:
        i. Title
        ii. Desc
        iii. Category Name
        iv. Release Year
        v. Language Name
        vi. Length
        vii. Rating
        viii. Actor Names (array)

2. List API will allow to list:
    a. Categories
    b. Languages
    c. Actors

3. Search log table - read client task and think which API is needed.

Main test - Client Side Task

1. Dashboard page will show:
    a. Search form
    b. Results table
    c. Search log table
2. Search form
    a. Will allow filtering by (any combination):
        i. Title
        ii. Description
        iii. Category Name
        iv. Actor name
        v. Language Name
    b. Will display results as you type (or edit filter)
    c. The filtering will allow free-text for all filter fields.
    d. The following fields will also have offer selection for filtering from list of available values:
        i. Categories
        ii. Languages
        iii. Actors
3. Results table
    a. Would display all results in following columns:
        i. Title
        ii. Desc
        iii. Category Name
        iv. Release Year
        v. Language Name
        vi. Length
        vii. Rating
        viii. Actor Names (concatenated)
    b. Would display results in pages (number of results in page 50)
4. Search log table
    Would display the audit lod for the search log with the values:
        a. Timestamp
        b. Search types (actor,title,desc,category…)
        c. Search values
        d. Result count